Randomized and Quantum Solution of Initial-Value Problems for Ordinary Differential Equations of Order k*

Marek Szczęsny

Faculty of Applied Mathematics, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Cracow, Poland,
E-mail: szczesny@uci.agh.edu.pl

Abstract

We study possible advantages of randomized and quantum computing over deterministic computing for scalar initial-value problems for ordinary differential equations of order k. For systems of equations of the first order this question has been settled modulo some details in [5]. A speed-up over deterministic computing shown in [5] is related to the increased regularity of the solution with respect to that of the right-hand side function. For a scalar equation of order k (which can be transformed into a special system of the first order), the regularity of the solution is increased by k orders of magnitude. This leads to improved complexity bounds depending on k for linear information in the deterministic setting, see [8]. This may suggest that in the randomized and quantum settings a speed-up can also be achieved depending on k.

We show in this paper that a speed-up dependent on k is not possible in the randomized and quantum settings. We establish lower complexity bounds, showing that the randomized and quantum complexities remain at the some level as for systems of the first order, no matter how large k is. Thus, the algorithms from [5] remain (almost) optimal, even if we restrict ourselves to a subclass of systems arising from scalar equations of order k.

Key words: k-th order initial-value problems, randomized computing, quantum computing, complexity.

1 Introduction

In the previous paper [8] we established complexity bounds for scalar initial-value problems for ordinary differential equations of order k in the deterministic worst-case setting. We showed for instance that if the right-hand side function g depends only on

* This research was partly supported by AGH grant No. 10.420.03

the solution function (not on the derivatives of the solution), then the ε-complexity for linear information is equal to $\Theta\left((1/\varepsilon)^{1/(r+k)}\right)$, where r denotes the regularity of g. Hence, the order k of the equation contributes significantly to the ε-complexity. In this paper, we study the dependence of the ε-complexity on k in the randomized and quantum settings. So far, the complexity in the randomized and quantum settings was considered for systems of equations of the first order, see [5]. For right-hand side functions with r continuous derivatives, the ε-complexity is essentially of order $(1/\varepsilon)^{1/(r+\varphi)}$, where $\varphi = 0$ in the deterministic setting, $\varphi = 1/2$ in the randomized setting, and $\varphi = 1$ in the quantum setting (for details see Section 3). A speed-up for systems of equations over deterministic computing is thus by $1/2$ in the exponent in the randomized case, and by 1 in the quantum case. Since a scalar equation of order k can be written as a special system of $k+1$ equations of the first order, the upper bounds from [5] remain valid in this case.

Intuitively, for scalar equations of order k, due to the increased regularity of the solution, one might expect a better speed-up over deterministic computing related to k. As we mentioned above, an improvement dependent on k is achieved in the deterministic worst-case setting by passing from the standard to linear (integral) information. One may hope that proper randomized or quantum approximation of the integrals involved in the computations will lead to algorithms with improved error bounds dependent on k.

We show in this paper that such an improvement is not possible, and a speed-up in the randomized and quantum settings is independent of k. We establish lower complexity bounds, showing that the complexity is $\Omega\left((1/\varepsilon)^{1/(r+1/2)}\right)$ in the randomized setting, and $\Omega\left((1/\varepsilon)^{1/(r+1)}\right)$ in the quantum setting, no matter how large k is. Hence, the algorithms defined in [5] remain almost optimal even if we restrict ourselves to special systems arising from scalar problems of order k.

The paper is organized as follows. We first introduce necessary definitions in the three settings. Then we recall for further comparison known complexity bounds for initial-value problems. Main results, lower bounds on the ε-complexity in the randomized and quantum settings, are shown in Section 4.

2 Problem definition

We consider the complexity of a problem in the following form

$$\begin{cases} u^{(k)}(x) = g\left(x, u(x), u'(x), \ldots, u^{(q)}(x)\right), & x \in [a,b], \\ u^{(j)}(a) = u_a^j, \quad j = 0, 1, \ldots, k-1, \end{cases} \tag{1}$$

where $0 \leq q < k$, $g : [a,b] \times \mathbb{R}^{q+1} \to \mathbb{R}$, $u : [a,b] \to \mathbb{R}$ $(a < b)$.
For $r \geqslant 1$ and given positive numbers $D_0, \ldots, D_r$, we consider the class of right-hand

side functions g defined by

$$\mathcal{G}^r = \big\{ g \mid g \in C^{(r)}([a,b] \times \mathbb{R}^{q+1}), \quad |\partial^j g(x,y)| \leqslant D_j, \quad \text{for} \\ x \in [a,b], \quad y \in \mathbb{R}^{q+1}, \quad j = 0,1,\ldots r \big\}, \tag{2}$$

where $\partial^j g$ represents all partial derivatives of order j of g.

Instead of (1) we can write an equivalent system of differential equations of order 1 of the form:

$$\mathbf{u}'(x) = \begin{bmatrix} u_0'(x) \\ u_1'(x) \\ \vdots \\ u_{k-1}'(x) \\ u_k'(x) \end{bmatrix} = \begin{bmatrix} 1 \\ u_2(x) \\ \vdots \\ u_k(x) \\ g(u_0(x), u_1(x), \ldots, u_{q+1}(x)) \end{bmatrix} = \mathbf{g}(\mathbf{u}(x)), \ x \in [a,b], \tag{3}$$

with a initial conditions

$$\mathbf{u}(a) = \left[a, u_a^0, \ldots, u_a^{k-1}\right]^T. \tag{4}$$

Then the solution $u(x)$ of (1) corresponds to the function $u_1(x)$.

Before we start analyzing the complexity of (1), we remind basic notions connected with deterministic, randomized and quantum computations. We are interested in finding a bounded function $l = l(x)$ that approximates the solution of (1). The construction of l is based on a certain information about the right-hand side function g. In the deterministic setting, we usually consider standard information, in which we compute values of g or its partial derivatives at some points, or linear information in which we know values of linear functionals of g.

In the randomized setting the values of g or its partial derivatives can be computed at randomly chosen points. In the quantum setting, information about g is gathered by applications of a quantum query for g. The reader is referred to [3] for a detailed explanation what a quantum query is.

To get an approximate solution $l(x)$, we use an algorithm A, which is a mapping from the information space into the space of bounded functions.
In the deterministic setting, the worst-case error of an algorithm A in class $\mathcal{G}^r$ is defined by

$$e^{\text{worst}}(A, \mathcal{G}^r) = \sup_{g \in \mathcal{G}^r} \sup_{x \in [a,b]} |u(x) - l(x)|. \tag{5}$$

In the randomized and quantum settings an approximation obtained is random. Letting $(\Omega, \Sigma, \mathbf{P})$ be a probability space, an algorithm A provides us with an approximate solution l^ω, where $\omega \in \Omega$.
The error of the algorithm A at g is defined by

$$e^\omega(A, g) = \sup_{x \in [a,b]} |u(x) - l^\omega(x)| \tag{6}$$

(we assume that $e^\omega(A,g)$ is a random variable for each $g \in \mathcal{G}^r$). The error of A in the class $\mathcal{G}^r$ in the randomized setting is defined by

$$e^{\text{rand}}(A, \mathcal{G}^r) = \sup_{g \in \mathcal{G}^r} \left(\mathbf{E}(e^\omega(A,g))^2\right)^{1/2}, \tag{7}$$

and in the quantum setting by

$$e^{\mathrm{quant}}(A,\mathcal{G}^r) = e^{\mathrm{quant}}(A,\mathcal{G}^r,\delta) = \sup_{g\in\mathcal{G}^r} \inf\{\alpha \mid \mathbf{P}(e^{\omega}(A,g) > \alpha) \le \delta\}. \tag{8}$$

The number $\delta \in (0,1/2)$ denotes here the failure probability. It is often assumed to be $1/4$. The success probability can then be increased by taking a median of a number of repetitions of an algorithm A (see [3]).

By the cost in the deterministic, randomized and quantum settings, we mean a number of subroutine calls for g, which is used by an algorithm A. Thus, in the deterministic and randomized settings, the cost is equal to number of evaluations of g or its partial derivatives, while in the quantum setting it is a number of quantum query calls. We will denote the cost in the respective setting by $\mathrm{cost}^{\mathrm{worst}}(A)$, $\mathrm{cost}^{\mathrm{rand}}(A)$ or $\mathrm{cost}^{\mathrm{quant}}(A)$.

For any $\varepsilon > 0$, by the ε-complexity of the problem we mean the minimal cost sufficient to solve the problem with error no larger than ε, where the minimum is taken over all algorithms solving the problem

$$\mathrm{comp}(\mathcal{G}^r,\varepsilon) \;=\; \min_{A} \{\mathrm{cost}(A) \mid e(A,\mathcal{G}^r) \leqslant \varepsilon\}. \tag{9}$$

To denote the complexity in the deterministic, randomized or quantum settings, we will use a suitable superscript: "worst", "rand" or "quant". Additionally to denote different types of information used in the deterministic setting, we will use indices: "worst-st" and "worst-lin" for standard and linear information, respectively.

3 Upper complexity bounds

In this section we briefly recall known complexity bounds for scalar equations of order k, as well as those for systems of the first order.

In [5], Kacewicz dealt with systems of equations of the first order of the form

$$z'(t) = f(z(t)), \quad t \in [a,b], \quad z(a) = \eta, \tag{10}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ and $\eta \in \mathbb{R}^d$. He considered the Hölder class of functions

$$\begin{aligned}\mathcal{F}^{r,\rho} = \big\{ f : \mathbb{R}^d \to \mathbb{R}^d \mid f \in C^{(r)}(\mathbb{R}^d), \quad |\partial^i f^j(y)| \le D_i,\ i = 0,1,\ldots,r,\\ |\partial^r f^j(y) - \partial^r f^j(z)| \le H\|y-z\|^{\rho},\ y,z \in \mathbb{R}^d, \quad j = 1,2,\ldots,d\big\},\end{aligned} \tag{11}$$

where $\rho \in (0,1]$.
It was shown in [5] that the ε-complexity is

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{F}^{r,\rho},\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+\rho+1/2-\gamma)}\right) \tag{12}$$

and

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{F}^{r,\rho},\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+\rho+1-\gamma)}\right) \tag{13}$$

with an arbitrarily small positive parameter γ. (The constants in the big-O notation depend on γ.) These bounds are almost optimal, i.e. they essentially match lower bounds on the complexity.

It is easy to see that the bounds above with $\rho = 0$ hold for the class $\mathcal{F}^r$, where

$$\mathcal{F}^r = \big\{ f : \mathbb{R}^d \to \mathbb{R}^d \mid f \in C^{(r)}(\mathbb{R}^d), \quad |\partial^i f^j(y)| \le D_i,\ i = 0, 1, \ldots, r, \\ y \in \mathbb{R}^d, \quad j = 1, 2, \ldots, d \big\}. \tag{14}$$

For systems (10) the ε-complexity in the class $\mathcal{F}^r$ is thus equal to

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{F}^r, \varepsilon) = O\left(\left(\frac{1}{\varepsilon} \right)^{1/(r+1/2-\gamma)} \right) \tag{15}$$

and

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{F}^r, \varepsilon) = O\left(\left(\frac{1}{\varepsilon} \right)^{1/(r+1-\gamma)} \right). \tag{16}$$

Since equation (1) can be transformed into a system of the first order (3), upper complexity bounds (15) and (16) are still valid for scalar equations of order k with $g \in \mathcal{G}^r$. (Although function $\mathbf{g}$ has unbounded components, we can consider an equivalent problem with a bounded right-hand side function having bounded partial derivatives up to the rth order.)

The question that we deal with in this paper is whether these bounds can be improved. We ask if a speed-up can be achieved dependent on k due to the increased regularity of the solution. In some cases in the deterministic worst-case setting such a speed-up dependent on k can indeed be shown. We have shown in [8] for standard information that

$$\mathrm{comp}^{\text{worst-st}}(\mathcal{G}^r, \varepsilon) = \Theta\left(\left(\frac{1}{\varepsilon} \right)^{1/r} \right), \tag{17}$$

so that there is no dependence on k in this case. However, if we allow linear information about right-hand side function, we can achieve a better result. The use of integral information leads (for $q = 0$) to the upper bound

$$\mathrm{comp}^{\text{worst-lin}}(\mathcal{G}^r, \varepsilon) = O\left(\left(\frac{1}{\varepsilon} \right)^{1/(r+k)} \right). \tag{18}$$

The complexity in this case significantly depends on k.

Intuitively, one may expect a speed-up dependent on k by replacing integrals in deterministic algorithms by effective randomized or quantum approximations. In the next section we show lower bounds on the complexity in both settings, which indicate that this intuition turns out not to be correct.

4 Lower complexity bounds in the randomized and quantum settings

We show in this section the main results of this paper. We prove the following lower bounds on randomized and quantum complexity of equations of order k with the right-hand side function belonging to class $\mathcal{G}^r$. Together with (15) and (16) they show that the complexity of scalar equations (1) is independent of k.

Theorem 4.1 *Let $r \geqslant 1$. For an arbitrary k*

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{G}^r, \varepsilon) = \Omega\left(\frac{1}{\varepsilon}\right)^{1/(r+1/2)}, \tag{19}$$

and

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{G}^r, \varepsilon) = \Omega\left(\frac{1}{\varepsilon}\right)^{1/(r+1)}, \tag{20}$$

where the constants in the "Ω" notation depend only on the class $\mathcal{G}^r$ and k.

Before proving Theorem 4.1 we give some auxiliary results. We shall consider a subclass $\mathcal{G}_1^r$ of the class $\mathcal{G}^r$ given by

$$\mathcal{G}_1^r = \{g : [a,b] \to \mathbb{R} \mid g \in C^{(r)}([a,b]) \sup_{x\in[a,b]} \left|g^{(j)}(x)\right| \leqslant D_j,\ j = 0, \ldots, r\}. \tag{21}$$

This class includes functions dependent on x only, that is, problem (1) for $g \in \mathcal{G}_1^r$ reduces to iterated integration. Since $\mathcal{G}_1^r \subset \mathcal{G}^r$, we have that $\mathrm{comp}(\mathcal{G}_1^r, \varepsilon) \leqslant \mathrm{comp}(\mathcal{G}^r, \varepsilon)$.

We now show properties of auxiliary functions that will be used in the proof of Theorem 4.1. Let

$$\psi_r(x) := \frac{1}{(r-1)!} \int_{-1}^{1} (x-t)_+^{r-1} \mathrm{sgn} U_r(t) \mathrm{d}t \tag{22}$$

be a perfect B-spline of degree r with r knots in $(-1, 1)$, see e.g. [1]. Function U_r is the Tchebycheff polynomial of second kind, and $(x-t)_+^{r-1}$ a truncated power function:

$$(x-t)_+^{r-1} := \begin{cases} (x-t)^{r-1} & \text{for} \quad x \geqslant t, \\ 0 & \text{for} \quad x < t. \end{cases} \tag{23}$$

For an interval $[c, d]$, we define by using $\psi_r(x)$ a function

$$\varphi_r([c,d], x) := \alpha \left(\frac{d-c}{2}\right)^r \psi_{r+1}\left(\frac{2}{d-c}x - \frac{d+c}{d-c}\right), \quad x \in [c,d], \tag{24}$$

where $m_j := \sup_{x\in[-1,1]} |\psi_{r+1}^{(j)}(x)|$ and $\alpha := \min_{j=0,1,\ldots,r} D_j/m_j$. From the properties of perfect B-spline, the function $\varphi_r([c,d], \cdot)$ belongs to the class $C^{(r)}([c,d])$, and $\varphi_r^{(j)}([c,d], c) =$

$\varphi_r^{(j)}([c,d],d) = 0$ for $j = 0,\dots,r$. In the sequel we will use the functions $\varphi_r([c,d],\cdot)$ on intervals $[c,d]$ such that $d-c<2$ and $[c,d]\subset[a,b]$. Note that $\varphi_r([c,d],\cdot)\in\mathcal{G}_1^r$ for $d-c<2$ (if we define $\varphi_r([c,d],\cdot)=0$ beyond the interval $[c,d]$).

In the proof of Theorem 4.1 we will need the value of the iterated integral of $\varphi_r([c,d],\cdot)$.

Fact 4.1 *Let $m\in\mathbb{N}$, $m\geq 1$ and $\varphi_r^m := \left(\frac{1}{2}\right)^{r+m}\frac{\alpha}{(r+m)!}\int_{-1}^{1}(1-t)^{r+m}\mathrm{sgn}\ U_{r+1}(t)\mathrm{d}t$.*

Then

$$\int_c^d\int_c^{y_{m-1}}\dots\int_c^{y_1}\varphi_r([c,d],y_0)\mathrm{d}y_0\,\mathrm{d}y_1\dots\mathrm{d}y_{m-1} = (d-c)^{r+m}\varphi_r^m. \tag{25}$$

Proof. Changing variables $x_i = (2y_i-d-c)/(d-c)$, $i=0,1,\dots,m-1$ in the left-hand side of (25), we get

$$\begin{aligned}
L &= \int_c^d\int_c^{y_{m-1}}\dots\int_c^{y_1}\varphi_r([c,d],y_0)\mathrm{d}y_0\,\mathrm{d}y_1\dots\mathrm{d}y_{m-1}\\
&= \alpha\left(\frac{d-c}{2}\right)^{r+m}\int_{-1}^{1}\int_{-1}^{x_{m-1}}\dots\int_{-1}^{x_1}\psi_{r+1}(x_0)\mathrm{d}x_0\,\mathrm{d}x_1\dots\mathrm{d}x_{m-1}\\
&= \alpha\left(\frac{d-c}{2}\right)^{r+m}\int_{-1}^{1}\int_{-1}^{x_{m-1}}\dots\int_{-1}^{x_1}\frac{1}{r!}\int_{-1}^{1}(x_0-t)_+^r\mathrm{sgn}U_{r+1}(t)\mathrm{d}t\,\mathrm{d}x_0\,\mathrm{d}x_1\dots\mathrm{d}x_{m-1}.
\end{aligned}$$

Changing the order of integration with respect to t and x_0, and computing next the inner integral, we get

$$L = \alpha\left(\frac{d-c}{2}\right)^{r+m}\int_{-1}^{1}\int_{-1}^{x_{m-1}}\dots\int_{-1}^{x_2}\frac{1}{(r+1)!}\int_{-1}^{1}(x_1-t)_+^{r+1}\mathrm{sgn}U_{r+1}(t)\mathrm{d}t\,\mathrm{d}x_1\dots\mathrm{d}x_{m-1}.$$

We proceed similarly for x_i, $i=1,\dots,m-1$ to get finally

$$\begin{aligned}
L &= \alpha\left(\frac{d-c}{2}\right)^{r+m}\frac{1}{(r+m)!}\int_{-1}^{1}(1-t)^{r+m}\mathrm{sgn}U_{r+1}(t)\mathrm{d}t\\
&= (d-c)^{r+m}\varphi_r^m.
\end{aligned}$$

□

For $n\in\mathbb{N}$, let $h := (b-a)/n$, and $a_i := a+ih$ for $i=0,1,\dots,n$. We now consider the following functions

$$f_i(x) := \begin{cases}\varphi_r([a_i,a_{i+1}],x) & \text{for} \quad x\in[a_i,a_{i+1}]\\ 0 & \text{for} \quad x\notin[a_i,a_{i+1}]\end{cases}, \quad \text{for} \quad i=0,1,\dots n-1. \tag{26}$$

Obviously, for $h < 2$ functions f_i belong to the class $\mathcal{G}_1^r$ (see the explanation after the definition of $\varphi_r([c,d],\cdot)$).

The following lemma shows how to express the k-fold integrals of the functions f_i.

Lemma 4.1 *Let functions f_i be defined as above. Then for any $k \geq 1$ and $n \geq 1$ we have*

$$\int_a^b \int_a^{y_{k-1}} \dots \int_a^{y_1} f_i(y_0)\mathrm{d}y_0\,\mathrm{d}y_1 \dots \mathrm{d}y_{k-1} = h^{r+k} \sum_{m=1}^{k} \frac{1}{(k-m)!}\varphi_r^m (n-i-1)^{k-m}. \tag{27}$$

Proof. To prove the lemma we shall show that

$$\int_a^x \int_a^{y_{k-1}} \dots \int_a^{y_1} f_i(y_0)\mathrm{d}y_0 \dots \mathrm{d}y_{k-2}\,\mathrm{d}y_{k-1} =$$

$$= \begin{cases} 0 & \text{for} \quad x \in [a, a_i) \\ \displaystyle\int_{a_i}^x \int_{a_i}^{y_{k-1}} \dots \int_{a_i}^{y_1} \varphi_r([a_i, a_{i+1}], y_0)\mathrm{d}y_0\,\mathrm{d}y_1 \dots \mathrm{d}y_{k-1} & \text{for} \quad x \in [a_i, a_{i+1}] \\ \displaystyle\sum_{m=1}^{k} \frac{h^{r+m}}{(k-m)!}\varphi_r^m (x - a_{i+1})^{k-m} & \text{for} \quad x \in (a_{i+1}, b] \end{cases}. \tag{28}$$

We prove this by induction with respect to k. For $k = 1$, we have from Fact 4.1 with $m = 1$ that

$$\int_a^x f_i(y)\mathrm{d}y = \begin{cases} 0 & \text{for} \quad x \in [a, a_i) \\ \displaystyle\int_{a_i}^x \varphi_r([a_i, a_{i+1}], y)\,\mathrm{d}y & \text{for} \quad x \in [a_i, a_{i+1}] \\ h^{r+1}\varphi_r^1 & \text{for} \quad x \in (a_{i+1}, b], \end{cases}$$

so that the statement holds true. Let us assume that (28) holds for $k-1$.
We consider three cases. By the definition of f_i we have

(i) for $x \in [a, a_i)$

$$\int_a^x \int_a^{y_{k-1}} \dots \int_a^{y_1} f_i(y_0)\,\mathrm{d}y_0 \dots \mathrm{d}y_{k-2}\,\mathrm{d}y_{k-1} = 0,$$

(ii) for $x \in [a_i, a_{i+1}]$

$$\int_a^x \int_a^{y_{k-1}} \dots \int_a^{y_1} f_i(y_0)\,\mathrm{d}y_0 \dots \mathrm{d}y_{k-2}\,\mathrm{d}y_{k-1} = \int_{a_i}^x \int_{a_i}^{y_{k-1}} \dots \int_{a_i}^{y_1} \varphi_r([a_i, a_{i+1}], y_0)\,\mathrm{d}y_0 \dots \mathrm{d}y_{k-2}\,\mathrm{d}y_{k-1}.$$

(iii) Let $x \in (a_{i+1}, b]$. We write

$$\int_a^x \int_a^{y_{k-1}} \dots \int_a^{y_1} f_i(y_0)\,\mathrm{d}y_0 \dots \mathrm{d}y_{k-2}\,\mathrm{d}y_{k-1}$$

$$= \int_{a_i}^{a_{i+1}} \int_{a_i}^{y_{k-1}} \dots \int_{a_i}^{y_1} \varphi_r([a_i, a_{i+1}], y_0)\,\mathrm{d}y_0 \dots \mathrm{d}y_{k-2}\,\mathrm{d}y_{k-1}$$

$$+ \int_{a_{i+1}}^{x} \sum_{m=1}^{k-1} \frac{h^{r+m}}{(k-1-m)!} \varphi_r^m (y_{k-1} - a_{i+1})^{k-1-m}\,\mathrm{d}y_{k-1}.$$

The last term in the equation above follows from the inductive assumption. Integrating the last term and using Fact 4.1 with $m = k$ to the first term, we get that

$$\int_a^x \int_a^{y_{k-1}} \dots \int_a^{y_1} f_i(y)\,\mathrm{d}y \dots \mathrm{d}y_{k-2}\,\mathrm{d}y_{k-1} = h^{r+k}\varphi_r^k + \sum_{m=1}^{k-1} \frac{h^{r+m}}{(k-m)!} \varphi_r^m (x - a_{i+1})^{k-m}.$$

This ends the inductive proof of (28). To prove Lemma 4.1, it is sufficient to take $x = b$ in (28).

□

We are ready to prove Theorem 4.1.

Proof of Theorem 4.1 We first prove a lower bound on the ε-complexity in the quantum setting. Consider the subclass $\mathcal{G}_1^r$ of $\mathcal{G}^r$. Let ϕ be any quantum algorithm solving problem (1) such that $e^{\mathrm{quant}}(\phi, \mathcal{G}^r) \le \varepsilon$ (this yields that $e^{\mathrm{quant}}(\phi, \mathcal{G}_1^r) \le \varepsilon$). We shall prove that $\mathrm{cost}^{\mathrm{quant}}(\phi) = \Omega\left((1/\varepsilon)^{1/(r+1)}\right)$.

Note that for any function $g \in \mathcal{G}_1^r$ the problem (1) reduces to the computation of the k-fold integral

$$u(x) = \sum_{j=0}^{k-1} \frac{u_a^j}{j!}(x-a)^j + \int_a^x \int_a^{t_{k-1}} \dots \int_a^{t_1} g\,(t_0)\,\mathrm{d}t_0\,\mathrm{d}t_1 \dots \mathrm{d}t_{k-1}, \qquad x \in [a,b]. \tag{29}$$

We prove the lower bound by showing that the solution of a finite number of problems (29) with suitable right-hand side functions leads to the solution of the mean value problem.

Let $n \in \mathbb{N}$. We divide the interval $[a, b]$ into $2kn$ parts, with points $a_i = a + i\bar{h}$, where $i = 0, 1, \dots, 2kn$ and $\bar{h} := (b-a)/(2kn)$. Let f_i be the functions from (26) with $n := 2kn$ and $h := \bar{h}$.

Let λ_i be arbitrary numbers with $|\lambda_i| \leqslant 1$ for $i = 0, \ldots, kn-1$, and let $\mathcal{X}$ be an injective mapping $\mathcal{X} : \{0, 1, \ldots, kn-1\} \to \{0, 1, \ldots, 2kn-1\}$. Then obviously $\sum_{i=0}^{kn-1} \lambda_i f_{\mathcal{X}(i)} \in \mathcal{G}_1^r$.

Consider k such mappings $\mathcal{X}_j$ and numbers c_j to be specified later on, $j = 0, 1, \ldots, k-1$. From Lemma 4.1, we have that

$$\begin{aligned} S &= \sum_{j=0}^{k-1} c_j \int_a^b \int_a^{y_{k-1}} \cdots \int_a^{y_1} \sum_{i=0}^{kn-1} \lambda_i f_{\mathcal{X}_j(i)}(y_0) \mathrm{d}y_0 \, \mathrm{d}y_1 \ldots \mathrm{d}y_{k-1} \\ &= \sum_{j=0}^{k-1} c_j \sum_{i=0}^{kn-1} \lambda_i \bar{h}^{r+k} \sum_{m=1}^{k} \frac{1}{(k-m)!} \varphi_r^m (2kn - \mathcal{X}_j(i) - 1)^{k-m}. \end{aligned}$$

After changing the summation order, we get that

$$S = \bar{h}^{r+1} \sum_{i=0}^{kn-1} \lambda_i \sum_{m=1}^{k} (b-a)^{k-m} \bar{h}^{m-1} \frac{1}{(k-m)!} \varphi_r^m \sum_{j=0}^{k-1} c_j \left(1 - \frac{\mathcal{X}_j(i)+1}{2kn}\right)^{k-m}.$$

We split the sum indexed by m into two parts. We have

$$\begin{aligned} S = \bar{h}^{r+1} &\sum_{i=0}^{kn-1} \lambda_i (b-a)^{k-1} \frac{1}{(k-1)!} \varphi_r^1 \sum_{j=0}^{k-1} c_j \left(1 - \frac{\mathcal{X}_j(i)+1}{2kn}\right)^{k-1} \\ &+ \bar{h}^{r+1} \sum_{i=0}^{kn-1} \lambda_i \sum_{m=2}^{k} (b-a)^{k-m} \bar{h}^{m-1} \frac{1}{(k-m)!} \varphi_r^m \sum_{j=0}^{k-1} c_j \left(1 - \frac{\mathcal{X}_j(i)+1}{2kn}\right)^{k-m}. \end{aligned}$$

We now show that there exist mappings $\mathcal{X}_j$ and numbers c_j such that for any i

$$\sum_{j=0}^{k-1} c_j \left(1 - \frac{\mathcal{X}_j(i)+1}{2kn}\right)^{k-1} = 1, \tag{30}$$

and for $m = 2, \ldots, k$,

$$\sum_{j=0}^{k-1} c_j \left(1 - \frac{\mathcal{X}_j(i)+1}{2kn}\right)^{k-m} = 0. \tag{31}$$

We now define the functions $\mathcal{X}_j$ for $j = 0, \ldots, k-1$. Let

$$\mathcal{X}_j(i) = n(k-j) + i - 1, \qquad i = 0, \ldots, kn-1. \tag{32}$$

Then obviously $\mathcal{X}_j(i) \in \{0, \ldots, 2kn-1\}$. For such $\mathcal{X}_j$ and $m = 1, \ldots, k$, we get that

$$\begin{aligned} \sum_{j=0}^{k-1} c_j \left(1 - \frac{\mathcal{X}_j(i)+1}{2kn}\right)^{k-m} &= \sum_{j=0}^{k-1} c_j \left(1 - \frac{n(k-j)+i}{2kn}\right)^{k-m} \\ &= \sum_{j=0}^{k-1} c_j \left(\frac{1}{2} + \frac{j}{2k} - \frac{i}{2kn}\right)^{k-m}. \end{aligned}$$

Let now

$$w(x) = \sum_{j=0}^{k-1} c_j \left(\frac{1}{2} + \frac{j}{2k} - x \right)^{k-1} \tag{33}$$

be a polynomial of x, $x \in \mathbb{R}$. We select c_j, $j = 0, \ldots, k-1$, to be numbers such that

$$w(x) \equiv 1. \tag{34}$$

The existence of such numbers follows from the fact that (34) defines a system of linear equations with unknown variables c_j and with a transpose Vandermonde matrix. The solution (numbers c_j) of this equation is independent of n. From (34) we have that

$$w^{(m)}(x) \equiv 0, \tag{35}$$

for $m = 1, \ldots, k-1$. Hence

$$\sum_{j=0}^{k-1} c_j \left(\frac{1}{2} + \frac{j}{2k} - x \right)^{k-m} \equiv 0 \tag{36}$$

for $m = 2, \ldots, k$. Taking $x = i/(2kn)$ in (34) and (36) we have the desired property

$$\sum_{j=0}^{k-1} c_j \left(\frac{1}{2} + \frac{j}{2k} - \frac{i}{2kn} \right)^{k-m} = \begin{cases} 1 & \text{for} \quad m = 1 \\ 0 & \text{for} \quad m = 2, \ldots, k-1 \end{cases}, \tag{37}$$

for all $i = 0, 1, \ldots, kn-1$. Consequently, (30) and (31) hold true, and

$$\begin{aligned} S = \sum_{j=0}^{k-1} c_j \int_a^b \int_a^{y_{k-1}} \cdots \int_a^{y_1} \sum_{i=0}^{kn-1} \lambda_i f_{\mathcal{X}_{j(i)}}(y_0) \mathrm{d}y_0 \, \mathrm{d}y_1 \ldots \mathrm{d}y_{k-1} \\ = \bar{h}^{r+1} (b-a)^{k-1} \frac{1}{(k-1)!} \varphi_r^1 \sum_{i=0}^{kn-1} \lambda_i \\ = (kn)^{-r} (b-a)^{k+r} 2^{-r-1} \frac{1}{(k-1)!} \varphi_r^1 \left(\frac{1}{kn} \sum_{i=0}^{kn-1} \lambda_i \right). \end{aligned} \tag{38}$$

Denoting

$$I_j = \int_a^b \int_a^{y_{k-1}} \cdots \int_a^{y_1} \sum_{i=0}^{kn-1} \lambda_i f_{\mathcal{X}_{j(i)}}(y_0) \mathrm{d}y_0 \ldots \mathrm{d}y_{k-1}, \tag{39}$$

we have from (38) that

$$\frac{1}{kn} \sum_{i=0}^{kn-1} \lambda_i = (kn)^r (b-a)^{-(k+r)} 2^{r+1} (k-1)! \frac{1}{\varphi_r^1} \sum_{j=0}^{k-1} c_j I_j. \tag{40}$$

For each j, we now use the algorithm ϕ to compute an approximation A_j to I_j with error at most ε and probability at least $3/4$. This is done with cost equal to $\mathrm{cost}^{\mathrm{quant}}(\phi)$. The algorithm ϕ approximates the solution $u(x)$ of problem (1), in particular it gives us an approximation of $u(b)$. Sience the function $u(x)$ satisfies (29) the algorithm ϕ can be used to approximate the k-fold integral. Repeating the algorithm $\Theta(\log k)$ times and computing the median, we improve the probability of success to be at least $(3/4)^{1/k}$. This is achieved with cost equal to $\Theta\left(\log k \cdot \mathrm{cost}^{\mathrm{quant}}(\phi)\right)$.
Denote $C = (b-a)^{-(k+r)} 2^{r+1} (k-1)!/\varphi_r^1$. It follows from (40) that the mean of λ_i's, $1/(kn) \sum\limits_{i=0}^{kn-1} \lambda_i$, is approximated by $(kn)^r C \sum\limits_{j=0}^{k-1} c_j A_j$ with error

$$\left| \frac{1}{kn} \sum_{i=0}^{kn-1} \lambda_i - (kn)^r C \sum_{j=0}^{k-1} c_j A_j \right| \le \varepsilon_1, \tag{41}$$

and probability at least $3/4$, where $\varepsilon_1 := (kn)^r C \sum\limits_{j=0}^{k-1} |c_j| \varepsilon$. The cost of doing this is $O\left(k \log k \cdot \mathrm{cost}^{\mathrm{quant}}(\phi)\right)$. We now use lower complexity bound of Nayak and Wu for computing the mean of kn numbers, see [7]. They showed that the cost of any algorithm for computing the mean must be $\Omega(\min\{kn, 1/\varepsilon_1\})$ quantum queries for $\lambda_0, \ldots, \lambda_{kn-1}$. Taking $n = \Theta\left((1/\varepsilon)^{1/(r+1)}\right)$ we have that $\Omega(\min\{kn, 1/\varepsilon_1\}) = \Omega\left((1/\varepsilon)^{1/(r+1)}\right)$, which gives us the desired lower bound. (Note that a query for $\lambda_0, \ldots, \lambda_{kn-1}$ is also a query for g of the form $g(x) = \sum\limits_{i=0}^{kn-1} \lambda_i f_{\mathcal{X}(i)}(x)$).

In the randomized setting we proceed in a similar way. The difference appears only in the part connected with computing the mean value by randomized algorithm. Algorithm ϕ gives us an approximation A_j of each integral I_j, for $j = 0, \ldots, k-1$, such that

$$\left(\mathbf{E} \left| A_j - I_j \right|^2 \right)^{1/2} \le \varepsilon. \tag{42}$$

The error of computing the mean value of λ's by using A_j is

$$\left(\mathbf{E} \left| \frac{1}{kn} \sum_{i=0}^{kn-1} \lambda_i - (kn)^r C \sum_{j=0}^{k-1} c_j A_j \right|^2 \right)^{1/2} = \left(\mathbf{E} \left| (kn)^r C \sum_{j=0}^{k-1} c_j \left(A_j - I_j \right) \right|^2 \right)^{1/2}. \tag{43}$$

Using the properties of the expectation and since

$$\left| \sum_{j=0}^{k-1} c_j \left(A_j - I_j \right) \right|^2 \le \frac{1}{2} \sum_{i,j=0}^{k-1} |c_i c_j| \left(\left| A_i - I_i \right|^2 + \left| A_j - I_j \right|^2 \right), \tag{44}$$

we get

$$\left(\mathbf{E}\left(\frac{1}{kn}\sum_{i=0}^{kn-1}\lambda_i-(kn)^rC\sum_{j=0}^{k-1}c_jA_j\right)^2\right)^{1/2}$$
$$\le (kn)^rC\left(\frac{1}{2}\sum_{i,j=0}^{k-1}|c_ic_j|\left(\mathbf{E}\,|A_i-I_i|^2+\mathbf{E}\,|A_j-I_j|^2\right)\right)^{1/2}\le\varepsilon_1, \quad (45)$$

where $\varepsilon_1=(kn)^rC\sum_{j=0}^{k-1}|c_j|\varepsilon$. It was shown in [6] that any approximation satisfying (45) must be based on $\Omega(\min\{kn,(1/\varepsilon_1)^2\})$ subroutine calls. Hence, to prove desired bound on the ε-complexity in the randomized setting it suffices to take $n=\Theta\left((1/\varepsilon)^{1/(r+1/2)}\right)$. This completes the proof of Theorem 4.1.

□

From bounds (15) and (16) and Theorem 4.1, we get the following summarizing corollary.

Corollary 4.1 *The complexity of initial-value problem (1) with an arbitrary $k\ge 1$ satisfies:*

- *in the randomized setting*

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{G}^r,\varepsilon)=O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1/2-\gamma)}\right) \quad (46)$$

and

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{G}^r,\varepsilon)=\Omega\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1/2)}\right), \quad (47)$$

- *in the quantum setting*

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{G}^r,\varepsilon)=O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1-\gamma)}\right) \quad (48)$$

and

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{G}^r,\varepsilon)=\Omega\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1)}\right), \quad (49)$$

where the positive constant γ is arbitrary small, and the constants in the big-O notation depend on γ.

It follows from the Corollary that the algorithms from [5] remain optimal in a subclass of systems arising from scalar equations of order k, for an arbitrary $k\ge 1$.

5 Final remarks

We showed in this paper some limitations of randomized and quantum computing. The intuition that the order k of a differential equation may help in randomized and quantum setting (which is the case in the deterministic setting with linear information) turned out not to be true. Randomized computation gives us a speed-up only by 1/2, and quantum computation by 1 independently of k, which is the same as in case of the first order equations. Thus, to optimally solve the special initial value problem (1), we can transform it to system (3) and apply algorithms for a general system of order 1 presented in [5].

Acknowledgement

I thank Bolesław Kacewicz for his valuable comments and suggestions.

References

[1] B. D. Bojanov, H. A. Hakopian, A. A. Sahakian, Spline Functions and Multivariate Interpolations, *Kluwer Academic Publishers, London* (1993)

[2] S. Heinrich and E. Novak, Optimal summation and integration by deterministic, randomized, and quantum algorithms, in K.T. Fang, F. J. Hickernell, H. Niederreiter (Eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, Springer Verlag, Berlin 2002, 50-62; see also http://arXiv.org/abs/quant-ph/0105114

[3] S. Heinrich, Quantum Summation with an Application to Integration, *J. Complexity* **18** (2002) 1-50; see also http://arXiv.org/abs/quant-ph/0105116

[4] B. Kacewicz, Randomized and quantum algorithms yield a speed-up for initial-value problems, *J. Complexity* **20** (2004) 821-834; see also http://arxiv.org/abs/quant-ph/0311148

[5] B. Kacewicz, Almost Optimal Solution of Initial-Value Problems by Randomized and Quantum Algorithms, *J. Complexity* **22** (2006) 676-690, see also http://arxiv.org/abs/quant-ph/0510045

[6] P. Mathé, The optimal error of Monte Carlo integration, *J. Complexity* **11** (1995) 394-415

[7] A. Nayak, F. Wu, (1999), The quantum query complexity of approximating the median and related statistics, *STOC*, May 1999, 384-393; see also http://arXiv.org/abs/quant-ph/9804066

[8] M. Szczęsny, Complexity of Initial-Value Problems for Ordinary Differential Equations of Order k, *J. Complexity* **22** (2006) 514-532